

WHAT IS CLAIMED IS:

Sub A2

1. A processor including a memory, a plurality of execution units coupled to the memory and an array prefetch apparatus for transferring array data from the memory to the plurality of execution units in the processor, the array prefetch apparatus comprising:

2. an array prefetch queue coupled to the memory for receiving array data;

3. a first array prefetch queue pointer coupled to the array prefetch queue for designating in the array prefetch queue a location for loading the array data;

4. a second array prefetch queue pointer coupled to the array prefetch queue for designating in the array prefetch queue a location for accessing the array data;

5. an array prefetch controller coupled to the array prefetch queue and the first and second array prefetch queue pointers, the array prefetch controller for executing a load operation as an array load operation and an array move operation, the array load operation for accessing the array data from the memory and transferring the array data to the array prefetch queue at the location designated by the first pointer, the array move operation for moving the array data from the array prefetch queue at the location designated by the second pointer for accessing by the execution units of the processor.

1. 2. A processor according to Claim 1 wherein the array load operation and the array move operation are executed independently and asynchronously.

1. 3. A processor according to Claim 1 further comprising:
2. an array prefetch flag register coupled to the array prefetch controller, the array prefetch flag selectively directing the array prefetch controller to
3. execute the load operation as the array load operation and the array move operation for a first array prefetch flag register setting and to
4. execute the load operation as a load operation for a second array
5. prefetch flag register setting.

1 4. A processor according to Claim 1 wherein the array load operation inherits
2 attributes of the load operation that concern issuing of a memory request and
3 advancing of a memory address and wherein the array move operation inherits
4 attributes of the load operation that concern loading of a destination register by a read
5 data.

1 5. A processor according to Claim 1 further comprising a loop control logic
2 supporting software pipelining of loops, the loop control logic for executing a plurality
3 of stages (S) in a compiled, pipelined loop schedule of T cycles having an iteration
4 interval I, in which the loop control logic dynamically controls the number of stages in
5 an iteration as a function of the latencies of memory read operations.

Step 3/3

1 6. A processor according to Claim 1, further comprising:
2 a loop control logic supporting software pipelining of loops in a horizontal
3 processor, the loop control logic including:
4 a loop mode flag indicative of a current loop mode status, the loop
5 mode flag being set when a loop is executed;
6 a loop counter indicative of a first remaining number of logical
7 iterations in the loop being executed;
8 a prologue counter indicative of a second remaining number of logical
9 iterations in a prologue portion of the loop being executed; and
10 first enabling/disabling logic coupled to the loop mode flag and to the
11 prologue counter, the first enabling/disabling logic disabling
12 execution of operations in a first class of operations having side
13 effects.

5

1 7. A processor according to Claim 1, wherein the array prefetch queue further
2 comprises:
3 an array prefetch queue data memory; and
4 an array prefetch queue for valid bits.

6

1 8. A processor according to Claim 1, further comprising:

2 a plurality of array access channels, wherein the array prefetch queue further
3 comprises:

4 an array prefetch queue data memory including a plurality of channels,
5 the channels of the array prefetch data memory corresponding
6 one-to-one to the array access channels; and
7 an array prefetch queue for valid bits including a plurality of channels,
8 the channels of the array prefetch queue for valid bits
9 corresponding one-to-one to the array access channels.

15

1 A processor including an array prefetch apparatus for transferring array data
2 from a memory to a register, the array prefetch apparatus comprising:
3 an array prefetch queue coupled to the memory for receiving the array data;
4 an array prefetch queue tail pointer coupled to the array prefetch queue for
5 designating in the array prefetch queue a location for loading the array
6 data;
7 an array prefetch queue head pointer coupled to the array prefetch queue for
8 designating in the array prefetch queue a location for accessing the
9 array data and moving the array data to a register;
10 an array prefetch flag;
11 an array prefetch controller coupled to the array prefetch queue, the array
12 prefetch flag and the first and second array prefetch queue pointers, the array
13 prefetch controller for executing a load operation as a load
14 operation for a first setting of the array prefetch flag and alternatively,
15 for a second setting of the array prefetch flag, executing a load
16 operation as a combination of an array load operation and an array
17 move operation, the array load operation for accessing the array data
18 from the memory and transferring the array data to the array prefetch
19 queue at the location designated by the array prefetch queue tail
20 pointer, the array move operation for moving the array data from the
21 array prefetch queue at the location designated by the array prefetch
22 head pointer to a register designated by the array move operation.

40

16

15

1 10. A processor according to Claim 9, wherein the array load operation inherits
2 attributes of the load operation that concern issuing of a memory request and
3 advancing of a memory address and wherein the array move operation inherits
4 attributes of the load operation that concern loading of a destination register by a read
5 data.

17

15

1 11. A processor according to Claim 9 wherein the array load operation and the
2 array move operation are executed independently and asynchronously.

1 12. A processor according to Claim 9 further comprising a loop control logic
2 supporting software pipelining of loops, the loop control logic for executing a plurality
3 of stages (S) in a compiled, pipelined loop schedule of T cycles having an iteration
4 interval I, in which the loop control logic dynamically controls the number of stages in
5 an iteration as a function of the latencies of memory read operations.

1 13. A processor according to Claim 9, further comprising:
2 a loop control logic supporting software pipelining of loops in a horizontal
3 processor, the loop control logic including:
4 a loop mode flag indicative of a current loop mode status, the loop
5 mode flag being set when a loop is executed;
6 a loop counter indicative of a first remaining number of logical
7 iterations in the loop being executed;
8 a prologue counter indicative of a second remaining number of physical
9 iterations in a prologue portion of the loop being executed; and
10 first enabling/disabling logic coupled to the loop mode flag and to the
11 prologue counter, the first enabling/disabling logic disabling
12 execution of operations in a first class of operations having side
13 effects.

14. A method of transferring array data from a memory to a register

2 comprising the steps of:

3 designating in an array prefetch queue a location for loading array data;

44

4 designating in the array prefetch queue a location for accessing the array data
5 and moving the array data to a register;
6 executing a load operation as a combination of an array load operation and an
7 array move operation;
8 for the array load operation, accessing the array data from the memory and
9 transferring the array data to the array prefetch queue at the location for
10 loading array data;
11 for the array move operation, moving the array data from the array prefetch
12 queue at the location designated by the second pointer to a register
13 designated by the array move operation.

8 7
1 15. A method according to Claim 14 further comprising:
2 executing a load operation as a combination of an array load operation and an
3 array move operation for a first setting of an array prefetch flag; and
4 alternatively executing a load operation as a load operation for a second setting
5 of an array prefetch flag.

9 7
1 16. A method according to Claim 14 wherein the array load operation and the
2 array move operation are executed independently and asynchronously.

10 7
1 17. A method according to Claim 14, wherein the array load operation inherits
2 attributes of the load operation that concern issuing of a memory request and
3 advancing of a memory address and wherein the array move operation inherits
4 attributes of the load operation that concern loading of a destination register by a read
5 data.

11
1 18. A method of providing a processor including an array prefetch apparatus
2 for transferring array data from a memory to a register, the array prefetch apparatus
3 comprising the steps of:
4 providing an array prefetch queue coupled to the memory for receiving the
5 array data;

6 providing an array prefetch queue tail pointer coupled to the array prefetch
7 queue for designating in the array prefetch queue a location for loading
8 the array data;
9 providing an array prefetch queue head pointer coupled to the array prefetch
10 queue for designating in the array prefetch queue a location for
11 accessing the array data and moving the array data to a register;
12 providing an array prefetch flag;
13 providing an array prefetch controller coupled to the array prefetch queue, the
14 array prefetch flag and the first and second array prefetch queue
15 pointers, the array prefetch controller for executing a load operation as
16 a load operation for a first setting of the array prefetch flag and
17 alternatively, for a second setting of the array prefetch flag, executing a
18 load operation as a combination of an array load operation and an array
19 move operation, the array load operation for accessing the array data
20 from the memory and transferring the array data to the array prefetch
21 queue at the location designated by the array prefetch queue tail
22 pointer, the array move operation for moving the array data from the
23 array prefetch queue at the location designated by the array prefetch
24 head pointer to a register designated by the array move operation.

12 11
1 19. A method according to Claim 18, wherein the array load operation inherits
2 attributes of the load operation that concern issuing of a memory request and
3 advancing of a memory address and wherein the array move operation inherits
4 attributes of the load operation that concern loading of a destination register by a read
5 data.

13 11
1 20. A method according to Claim 18, wherein the array load operation and the
2 array move operation are executed independently and asynchronously.

